

proposed amendment cannot entered because the
 third limitation is not clear
Amendments to the Claims: whether the 1st process is the
 owner itself. Need to clarify
 this language

Claims 1-40 (Cancelled).

41. (Currently amended) A method for providing mutual exclusion for a resource in a computer system having a plurality of processes, the method comprising:

① maintaining a resource lock for each process requiring access to the resource, the resource lock having a plurality of fields requiring initialization in order for the process to access the resource, the plurality of fields including an owner indicator field for indicating an owner process for the resource;

② receiving, by a first process, an inquiry from a second process inquiring whether the first process owns the resource;

③ determining, by the first process, an owner process for the resource other than the first process; and

④ creating a ghost lock for the first process, wherein the ghost lock is a partial instantiation of a resource lock having at least the owner indicator field initialized to indicate the owner process determined for the resource but having less than all fields initialized, and wherein the ghost lock is maintained to facilitate future access to the resource by the first process allows the first process to identify the owner process for the resource without first sending an inquiry message to determine the owner process.

42. (Previously presented) The method of claim 41, wherein determining the owner process by the first process comprises:

determining that the first process is not the owner process; and
 determining thereby that the second process is the owner process.

43. (Previously presented) The method of claim 42, further comprising:

sending, by the first process, a response to the second process indicating that the first process is not the owner process for the resource; and

creating an owner lock for the second process, wherein the owner lock is a resource lock having all fields initialized and the owner indicator field indicating that the second process is the owner process for the resource.

44. (Previously presented) The method of claim 41, further comprising:

determining, by the second process, the owner process for the resource, the owner processing being one of the second process and a third process;

creating an owner lock for the second process if the second process is the owner process for the resource, wherein the owner lock is a resource lock having all fields initialized and the owner indicator field indicating that the second process is the owner process for the resource; and

creating a reference lock for the second process if the third process is the owner process for the resource, wherein the reference lock is a resource lock having all fields initialized and the owner indicator field indicating that the third process is the owner process for the resource.

45. (Previously presented) The method of claim 44, wherein determining the owner process by the second process comprises:

sending, by the second process, an inquiry to the third process inquiring whether the third process owns the resource;

receiving, by the second process, a response from the third process indicating whether the third process is the owner process for the resource; and

determining, by the second process, that the second process is the owner process for the resource, if the response indicates that the third process is not the owner process for the resource.

46. (Previously presented) The method of claim 45, further comprising:

sending, by the second process, an owner notification message to the first process indicating the owner process for the resource, the owner process being one of the second process and the third process.

47. (Previously presented) The method of claim 46, wherein determining the owner process by the first process comprises:

determining the owner process for the resource based upon the owner notification message.

48. (Previously presented) The method of claim 41, further comprising:
determining that the first process requires access to the resource;
identifying, by the first process, the owner process for the resource using the ghost lock; and

sending, by the first process, a request message to the owner process requesting access to the resource without first sending an inquiry message to determine the owner process.

49. (Previously presented) The method of claim 48, wherein identifying the owner process for the resource using the ghost lock comprises:

finding the ghost lock among a plurality of resource locks based upon a resource identifier; and

obtaining the owner process from the owner indicator field of the ghost lock.

50. (Previously presented) The method of claim 48, further comprising:
converting the ghost lock to a reference lock by initializing all uninitialized fields of the lock.

51. (Currently amended) An apparatus for providing mutual exclusion for a resource in a computer system, the device comprising:

a lock storage for storing resource locks, each resource lock having a plurality of fields requiring initialization in order for a corresponding process to access the resource, the plurality of fields including an owner indicator field for indicating an owner process for the resource; and

a distributed lock service process for managing resources locks in the lock storage, wherein the distributed lock service process is operably coupled to determine an owner process for the resource other than the distributed lock service process upon receiving an inquiry from another process inquiring whether the distributed lock service process owns the resource and to create a ghost lock in the lock storage, wherein the ghost lock is a partial instantiation of a resource lock having at least the owner indicator field initialized to indicate the owner process determined for the resource but having less than all fields initialized, and wherein ~~the distributed lock service process maintains the ghost lock to facilitate future access to the resource~~ the ghost lock allows the distributed lock service process to identify the owner process for the resource without first sending an inquiry message to determine the owner process.

52. (Previously presented) The apparatus of claim 51, wherein the distributed lock service process is operably coupled to determine that the other process is the owner process for the resource if the distributed lock service process is not the owner process for the resource.

53. (Previously presented) The apparatus of claim 51, wherein the distributed lock service process is operably coupled to send a response to the second process indicating that the first process is not the owner process for the resource.

54. (Previously presented) The apparatus of claim 51, wherein the distributed lock service process is operably coupled to determine the owner process for the resource based upon an owner notification message received

from the other process, the owner notification message identifying the owner process for the resource.

55. (Previously presented) The apparatus of claim 51, wherein the distributed lock service process is operably coupled to identify the owner process for the resource using the ghost lock upon requiring access to the resource and to send a request message to the owner process requesting access to the resource without first sending an inquiry message to determine the owner process.

56. (Previously presented) The apparatus of claim 55, wherein the distributed lock service process is operably coupled to identify the owner process for the resource using the ghost lock by finding the ghost lock among a plurality of resource locks in the loci storage based upon a resource identifier and obtaining the owner process from the owner indicator field of the ghost lock.

57. (Previously presented) The apparatus of claim 55, wherein the distributed lock service process is operably coupled to convert the ghost lock to a reference lock by initializing all uninitialized fields of the lock.

58. (Currently amended) An apparatus comprising a computer readable medium having embodied therein a computer program for providing mutual exclusion for a resource in a computer system, the computer program comprising:

a distributed lock service process for managing resources locks in a lock storage, each resource lock having a plurality of fields requiring initialization in order for a corresponding process to access the resource, the plurality of fields including an owner indicator field for indicating an owner process for the resource, wherein the distributed lock service process is programmed to determine an owner process for the resource other than the distributed lock service process upon receiving an inquiry from another process inquiring

whether the distributed lock service process owns the resource and to create a ghost lock in the lock storage, wherein the ghost lock is a partial instantiation of a resource lock having at least the owner indicator field initialized to indicate the owner process determined for the resource but having less than all fields initialized, and wherein ~~the distributed lock service process maintains the ghost lock to facilitate future access to the resource~~ the ghost lock allows the distributed lock service process to identify the owner process for the resource without first sending an inquiry message to determine the owner process.

59. (Previously presented) The apparatus of claim 58, wherein the distributed lock service process is programmed to determine that the other process is the owner process for the resource if the distributed lock service process is not the owner process for the resource.

60. (Previously presented) The apparatus of claim 58, wherein the distributed lock service process is programmed to send a response to the second process indicating that the first process is not the owner process for the resource.

61. (Previously presented) The apparatus of claim 58, wherein the distributed lock service process is programmed to determine the owner process for the resource based upon an owner notification message received from the other process, the owner notification message identifying the owner process for the resource.

62. (Previously presented) The apparatus of claim 58, wherein the distributed lock service process is programmed to identify the owner process for the resource using the ghost lock upon requiring access to the resource and to send a request message to the owner process requesting access to the resource without first sending an inquiry message to determine the owner process.

63. (Previously presented) The apparatus of claim 62, wherein the distributed lock service process is programmed to identify the owner process for the resource using the ghost lock by finding the ghost lock among a plurality of resource locks in the lock storage based upon a resource identifier and obtaining the owner process from the owner indicator field of the ghost lock.

64. (Previously presented) The apparatus of claim 62, wherein the distributed lock service process is programmed to convert the ghost lock to a reference lock by initializing all uninitialized fields of the lock.

65. (Currently amended) A computer system comprising a plurality of processes sharing a resource, wherein:

a resource lock is maintained for each process requiring access to the resource, the resource lock having a plurality of fields requiring initialization in order for the process to access the resource, the plurality of fields including an owner indicator field for indicating an owner process for the resource;

a first process receives an inquiry from a second process inquiring whether the first process owns the resource;

the first process determines an owner process for the resource other than the first process; and

a ghost lock is created for the first process, wherein the ghost lock is a partial instantiation of a resource lock having at least the owner indicator field initialized to indicate the owner process determined for the resource but having less than all fields initialized, and wherein the ghost lock ~~is maintained to facilitate future access to the resource by the first process~~ allows the first process to identify the owner process for the resource without first sending an inquiry message to determine the owner process.

66. (Previously presented) The computer system of claim 65, wherein the first process determines that the second process is the owner process for the resource if the first process is not the owner process for the resource.

67. (Previously presented) The computer system of claim 66, wherein:
the first process sends a response to the second process indicating that the first process is not the owner process for the resource; and
an owner lock is created for the second process, wherein the owner lock is a resource lock having all fields initialized and the owner indicator field indicating that the second process is the owner process for the resource.

68. (Previously presented) The computer system of claim 65, wherein:
the second process determines the owner process for the resource, the owner processing being one of the second process and a third process;
an owner lock is created for the second process if the second process is the owner process for the resource, wherein the owner lock is a resource lock having all fields initialized and the owner indicator field indicating that the second process is the owner process for the resource; and
a reference lock is created for the second process if the third process is the owner process for the resource, wherein the reference lock is a resource lock having all fields initialized and the owner indicator field indicating that the third process is the owner process for the resource.

69. (Previously presented) The computer system of claim 68, wherein the second process determines the owner process by:
sending an inquiry to the third process inquiring whether the third process owns the resource;
receiving a response from the third process indicating whether the third process is the owner process for the resource; and

determining that the second process is the owner process for the resource, if the response indicates that the third process is not the owner process for the resource.

70. (Previously presented) The computer system of claim 69, wherein: the second process sends an owner notification message to the first process indicating the owner process for the resource, the owner process being one of the second process and the third process.

71. (Previously presented) The computer system of claim 70, wherein the first process determines the owner process for the resource based upon the owner notification message.

72. (Previously presented) The computer system of claim 65, wherein: the first process identifies the owner process for the resource using the ghost lock upon requiring access to the resource and sends a request message to the owner process requesting access to the resource without first sending an inquiry message to determine the owner process.

73. (Previously presented) The computer system of claim 72, wherein the first process identifies the owner process using the ghost lock by finding the ghost lock among a plurality of resource locks based upon a resource identifier and obtaining the owner process from the owner indicator field of the ghost lock.

74. (Previously presented) The computer system of claim 72, wherein the first process converts the ghost lock to a reference lock by initializing all uninitialized fields of the lock.